

QuickStart

TracNav menu

- [News](#)
- [Download](#)
- [Install](#)
- [Getting Started](#)
- [Documentation](#)
- [Contribute](#)
- [Help](#)
- [Publications](#)
- [Testimonials](#)

The steps below should get you from just thinking about a configuration management system to an operational installation of Bcfg2. If you get stuck, be sure to check the [mailing list](#) or drop in to our [IRC channel](#).

Get & Install Bcfg2 Server

We recommend running the server on a Linux machine for ease of deployment due to the availability of packages for the dependencies.

First, you need to download and install Bcfg2. Our [Download](#) has both source and packages for common environments, while our [Install](#) page describes what to do once you have the packages in hand. To start you will need to install the server on one machine and the client on one or more machines. Yes, your server can also be a client (and should be by the time your environment is fully managed). Detailed installation instructions can be found on the [Install](#) page.

Set up Repository

The next step after installing the Bcfg packages is to configure the server. You can easily set up a personalized default configuration by running, on the server,

```
bcfg2-admin init
```

You will be presented with a series of questions that will build a Bcfg2 configuration file in `/etc/bcfg2.conf`, set up a skeleton repository (in `/var/lib/bcfg2` by default), help you create ssl certificates, and do any other similar tasks needed to get you started.

Once this process is done, you can start the Bcfg2 server:

```
/etc/init.d/bcfg2-server start
```

You can try it out by running the Bcfg2 client on the same machine, acting like it is your first client. **Note:** The following command will tell the client to run in no-op mode, meaning it will only check the client against the repository and report any changes it sees. It won't make any changes (partially because you haven't populated the repository with any yet). However, nobody is perfect - you can make a typo, our software can have bugs, monkeys can break in and hit enter before you are done. Don't run this command on a production system if you don't know what it does and aren't prepared for the consequences. We don't know of anybody having problems with it before, but it is

better to be safe than sorry. And now for the command:

```
bcfg2 -q -v -n
```

That can be translated as "bcfg2 quick verbose no-op". The output should be something similar to:

```
Loaded tool drivers:
  Chkconfig      POSIX          PostInstall  RPM

Phase: initial
Correct entries:      0
Incorrect entries:    0
Total managed entries: 0
Unmanaged entries:    242

Phase: final
Correct entries:      0
Incorrect entries:    0
Total managed entries: 0
Unmanaged entries:    242
```

Perfect! We have started out with an empty configuration, and none of our configuration elements are correct. It doesn't get much cleaner than that. But what about those unmanaged entries? Those are the extra configuration elements (probably all packages at the moment) that still aren't managed. Your goal now is to migrate each of those plus any it can't see up to the "Correct entries" line.

Populate Repository

Finally, you need to populate your repository. Unfortunately, from here on out we can't write up a simple recipe for you to follow to get this done. It is very dependent on your local configuration, your configuration management goals, the politics surrounding your particular machines, and many other similar details. We can, however, give you guidance.

After the above steps, you should have a toplevel repository structure that looks like:

```
bcfg-server:~ # ls /var/lib/bcfg2
Bundler/  Cfg/    Metadata/  Pkgmgr/  Rules/  SSHbase/  Svcmgr/  etc/
```

The place to start is the Metadata directory, which contains two files: `clients.xml` and `groups.xml`. Your current `clients.xml` will look pretty close to:

```
<Clients version="3.0">
  <Client profile="basic" pingable="Y" pingtime="0" name="bcfg-server.example.com"/>
</Clients>
```

The `clients.xml` file is just a series of `<Client />` tags, each of which describe one host you manage. Right now we only manage one host, the server machine we just created. This machine is bound to the `basic` profile, is pingable, has a pingtime of 0, and has the name `bcfg-server.example.com`. The two "ping" parameters don't matter to us at the moment, but the other two do. The `name` parameter is the fully qualified domain name of your host, and the `profile` parameter maps that host into the `groups.xml` file.

Our simple `groups.xml` file looks like:

```

<Groups version='3.0'>
  <Group profile='true' public='false' name='basic'>
    <Group name='suse' />
  </Group>
  <Group name='ubuntu' toolset='debian' />
  <Group name='debian' toolset='debian' />
  <Group name='redhat' toolset='rh' />
  <Group name='suse' toolset='rh' />
  <Group name='mandrake' toolset='rh' />
  <Group name='solaris' toolset='solaris' />
</Groups>

```

There are two types of groups in Bcfg: profile groups (`profile='true'`) and non-profile groups (`profile='false'`). Profile groups can act as top-level groups that can be clients can bind to, while non-profile groups only exist as members of other groups. In our simple starter case, we have a profile group named `basic`, and that is the group that our first client bound to. Our first client is a SuSE machine, so it contains the `suse` group. Of course, `bcfg2-admin` isn't smart enough to fill out the rest of your config, so the `suse` group further down is empty.

Let's say the first thing we want to set up on our machine is the message of the day. To do this, we simply need to create a Bundle and add that Bundle to an appropriate group. In this simple example, we start out by adding

```
<Bundle name='motd' />
```

to the `basic` group.

Next, we create a `motd.xml` file in the Bundler directory:

```

<Bundle name='motd' version='2.0'>
  <ConfigFile name='/etc/motd' />
</Bundle>

```

Now when we run the client, we get slightly different output:

```

Loaded tool drivers:
  Chkconfig    POSIX          PostInstall  RPM
Incomplete information for entry ConfigFile:/etc/motd; cannot verify

Phase: initial
Correct entries:      0
Incorrect entries:    1
Total managed entries: 1
Unmanaged entries:    242

In dryrun mode: suppressing entry installation for:
  ConfigFile:/etc/motd

Phase: final
Correct entries:      0
Incorrect entries:    1
Total managed entries: 1
Unmanaged entries:    242

```

We now have an extra unmanaged entry, bringing our total number of managed entries up to one. To manage it we need to copy `/etc/motd` to `/var/lib/bcfg2/Cfg/etc/motd/`. Note the layout of that path: all plain-text config files live in the `Cfg` directory. The directory structure under that directory directly mimics your real filesystem layout, making it easy to find and add new files. The last directory is the name of the file itself, so in this case the fill

path to the motd file would be `/var/lib/bcfg2/Cfg/etc/motd/motd`. Copy your real `/etc/motd` file to that location, run the client again, and you will find that we now have a correct entry:

```
Loaded tool drivers:
  Chkconfig    POSIX          PostInstall  RPM
```

```
Phase: initial
Correct entries:      1
Incorrect entries:    0
Total managed entries: 1
Unmanaged entries:    242
```

```
Phase: final
Correct entries:      1
Incorrect entries:    0
Total managed entries: 1
Unmanaged entries:    242
```

Done! Now we just have 242 (or more) entries to take care of!

The Bundler is a relatively easy directory to populate. You can find many samples of Bundles in the [BundleRepository](#), many of which can be used without editing.

Next Steps

Several other utilities can help from this point on:

`bcfg2-info` is a utility that instantiates a copy of the `bcfg2` server core (minus the networking code) for examination. From this, you can directly query:

- Client Metadata
- Which entries are provided by particular plugins
- Client Configurations

Run `bcfg2-info`, and type `help` and the prompt when it comes up.

`bcfg2-admin` can perform a variety of repository maintenance tasks. Run `bcfg2-admin help` for details.